Saskatchewan
Research Council

GRAPHICS SOFTWARE FOR DATA ANALYSIS:

II. DATA TRANSFER AND DATA FILE MANAGEMENT - USER'S GUIDE

by

J.W.J. Smith
and
M. Mellinger

SRC Technical Report No. 156

Publication R-851-2-B-84

June, 1984

GRAPHICS SOFTWARE FOR DATA ANALYSIS:

II. DATA TRANSFER AND DATA FILE MANAGEMENT - USER'S GUIDE

by

J.W.J. Smith
and
M. Mellinger

SRC Technical Report No. 156

The work reported here was carried out as part of the Geochemistry and Data Analysis project at the Saskatchewan Research Council.

This report is one from the series GRAPHICS SOFTWARE FOR DATA ANALYSIS. Information on other reports from this series may be obtained by inquiring to the:

Saskatchewan Research Council
30 Campus Drive
Saskatoon, Saskatchewan
S7N 0X1

June, 1984

SRC Publication No. R-851-2-B-84

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
	.1. Purpose	1
	.2. General Description	1
	.3. Hardware Configuration	1
	.4. Software Design	2
II.	GETTING STARTED	4
	.1. How to Use This Guide	4
	.2. How to Run The Programs	4
III.	PROGRAM OPERATION	5
	.1. COMPRESS Program	7
	a. Purpose	7
	b. Input	7
	c. Output	7
	d. Program operation	7
	.2. Data TRANSFER Program	15
	a. Purpose	15
	b. Input	15
	c. Output	15
	d. Program Operation	15
	.3. Data TRANSLATION Program	19
	a. Purpose	19
	b. Input	19
	c. Output	19

III. PROGRAM OPERATION (cont'd) Page

d. Program Operation 19

.4. SCANSTAT Program 25

a. Purpose 25

b. Input 25

c. Output 25

d. Program Operation 25

.5. CONTINGENCY TABLE Program 29

a. Purpose 29

b. Input 29

c. Output 29

d. Program Operation 29

.6. SUBFILE Program 33

a. Purpose 33

b. Input 33

c. Output 33

d. Program Operation 33

e. Example of Subfile Created 41

I. INTRODUCTION

I.1 PURPOSE

This guide provides a detailed description of the procedures used to: (1) prepare data files for transfer from the IBM 4341 mainframe computer (SASKCOMP) to the Tektronix 4051 microcomputer, (2) transfer the data from the former to the latter computer, and (3) translate the data on the Tektronix 4051 prior to further processing. Three programs which perform elementary statistics or data manipulation are also described.

I.2 GENERAL DESCRIPTION

Data stored in a disk data file on the IBM 4341 system are transferred to a Tektronix 4051 data file. Before the transfer, the data is compressed to eliminate useless information and to reduce the physical size of the transfer file. Transfer of the data file is then accomplished by connecting the Tektronix 4051 by leased telephone line to the IBM 4341. The data are transferred as ASCII characters and are later translated to binary codes in order to facilitate their use by the various Tektronix programs.

I.3 HARDWARE CONFIGURATION

The transmission operation requires: (1) the Tektronix 4051 and a local modem, and (2) a leased telephone line for access to the IBM 4341.

The translation of data from ASCII to binary, data manipulation and analysis is carried out on the Tektronix 4051. Data file storage and manipulation can be carried out using the internal magnetic tape drive only; for our present hardware, the ASCII file is initially

transferred to a tape file, and then translated to binary codes into a RAM-disk file.

The Tektronix 4051 used here has 32K of core memory, although the programs described in this report will run on 16K core memory. Output from the programs described here is on the graphics screen and on a line printer connected to the Tektronix 4051 via the ROM printer-interface.

The programs will also run on any upward-compatible hardware from the same manufacturer.

I.4 SOFTWARE DESIGN

The software described in this report is written to be interactive with the user in order to make both the transmission and manipulation of the data as clear and simple as possible.

The relationships between the programs covered in this report are shown in Figure 1.

A copy of the programs may be obtained on request by those who can access an IBM mainframe computer (COMPRESS program), or a Tektronix graphics computer of the 4050 series (other programs).

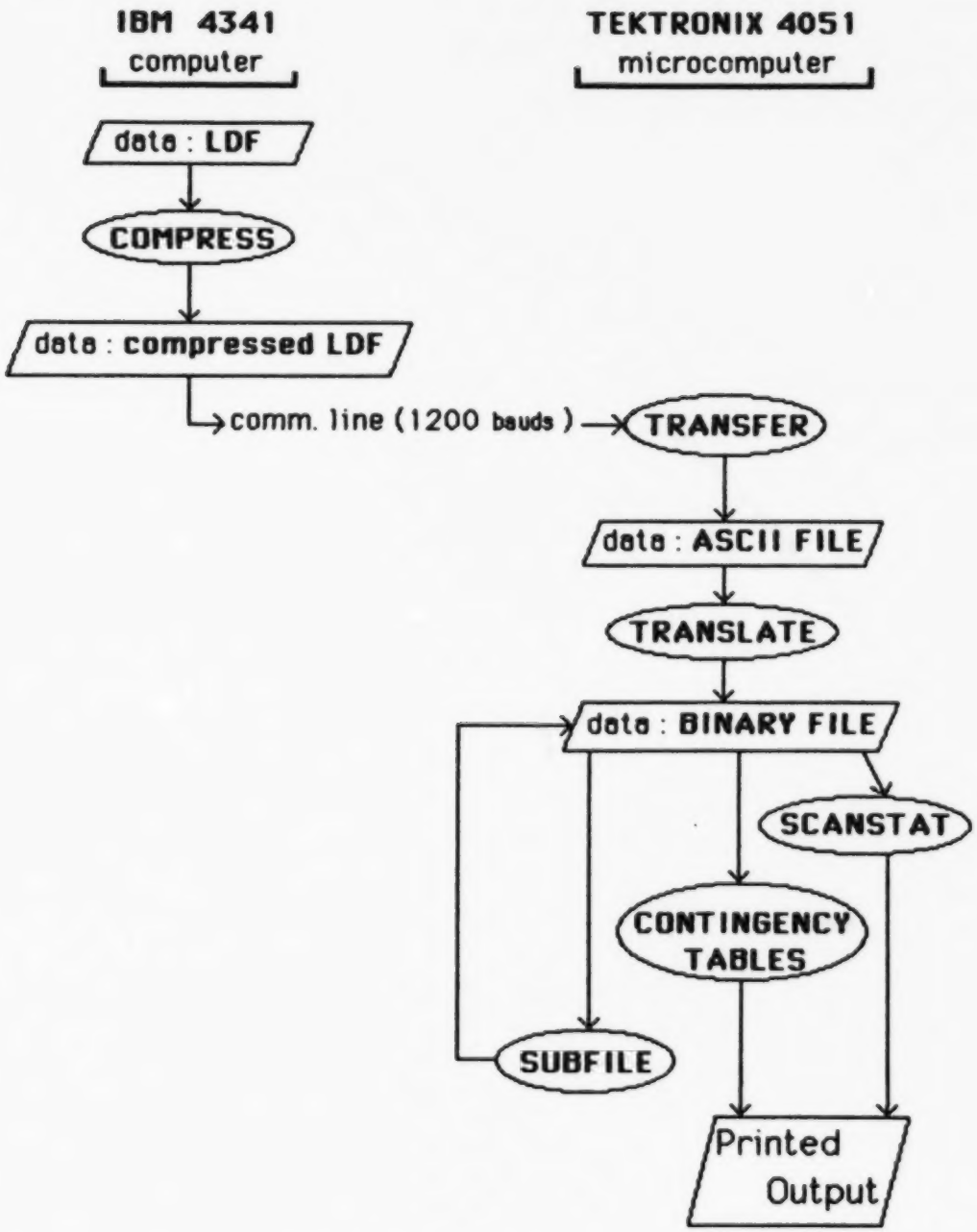

Figure 1. General relationships between the programs described in this report.

II. GETTING STARTED

II.1 HOW TO USE THIS GUIDE

This guide is composed of hard copies of the computer screen content or printout (left pages) and of one or more pages of explanations (right pages), all grouped in Chapter III.

Salient details of the screen contents are numbered, and explained successively; these numbers are circled, e.g.: (3). Computer-generated text is self-explanatory; the user's entries used as examples are boxed, e.g.: [30] .

II.2 HOW TO RUN THE PROGRAMS

In order to run the COMPRESS program, the user must be familiar with the TSO system of the IBM environment.

In order to run the other programs, and if the system must be started from power up, the user must be familiar enough with the Tektronix 4051 graphics computer to be able to load the selected program into core memory.

Once the selected program is in core memory, it is initialized by entering RUN from the keyboard. The program is now in control of the operations and will prompt the user as necessary, as explained in the appropriate section of chapter III.

III. PROGRAM OPERATION

The computer programs will be described in the order in which they would logically be used:

COMPRESS

TRANSFER

TRANSLATE

SCANSTAT

CONTINGENCY TABLES

SUBFILE

(Note: an explanation of the format and notations of Chapter III is given in Section II.1).

```
**
ENTER LOGON
LOGON       /
      LOGON IN PROGRESS AT 16:08:43 ON AUGUST 11, 1983
NO BROADCAST MESSAGES
READY

QED '       .TEST.DATA' OLD
DATA SET NOT LINE NUMBERED-NONUM ASSUMED
QED
VERIFY
TOP
83010501    77   7  10   4   8 444  21  15   2  87  14   4 3.8 0.4 0.5
INSERT
INPUT
ATHABASCA GEOCHEM DATA
   4
 LA KAOLCHLRILLT IR  AI CSEPANORMDEXCNOR
11111111111
1111111111111111
11100000001101000
11111110000000000

QED
LIST
ATHABASCA GEOCHEM DATA
   4
 LA KAOLCHLRILLT IR  AI CSEPANORMDEXCNOR
11111111111
1111111111111111
11100000001101000
11111110000000000
83010501    77   7  10   4   8 444  21  15   2  87  14   4 3.8 0.4 0.5
83010501     5   3  90                                10.01       1
83010501     9 100   0   0 888 888                5.8
QED
SAVE
SAVED
END
READY
```

(1) (2) (3) (4) (5)

III.1 COMPRESS PROGRAM

III.1.a Purpose: this is a FORTRAN 77 program which selects specified data from a Lithogeochemistry Data File (LDF), reduces the physical size of the file and prepares the file for transfer to the Tektronix 4051. The user is assumed to be familiar with the LDF format (Mellinger, 1983).*

III.1.b Input: a LDF stored in the IBM 4341 environment.

III.1.c Output: a disk data file in the IBM 4341 environment consisting of: data descriptors and a title, the compressed data records.

III.1.d Program Operation: The user connects, by cable, the Tektronix 4051 RS232 port to the modem and then accesses the IBM TSO system. Using QED, the user prefixes the existing LDF with a number of new lines (see further on left page for a summary).

(1) Title: up to 63 characters. It is suggested, however, to make the first 15 characters uniquely defined for each data file, due to later string display constraints.

(2) Number of records per sample (as defined in the LDF): using an I4 format.

In our example, each sample consists of 4 LDF records.

(3) Names of additional variables: these are for variables not pre-defined in the LDF, and which would appear in LDF records 04 and onward. The names must be coded in 17A4 format, in order to conform with the size of individual LDF records (17 fields per record).

(*) MELLINGER, M. (1983). The Lithogeochemistry Data File. User Manual. SRC Publication # R-740-1-E-83, 46 pages.

(III.1. COMPRESS ... cont'd)

In our example, with 4 LDF records per sample (see above), one must define the variable names of record 04 (variable names for records 01 to 03 inclusive are pre-defined LDF names): 10 names are defined, each occupying a 4-character field (LA, KAOL, CHLR, . . .).

(4) Variable selection codes: as many lines as there are LDF records per sample must be entered here, each line specifying the selection codes for one LDF record; input format is 12I1 for the first LDF record, and 17I1 for the subsequent records. Each selection code corresponds to one variable, and has for value either "1" (if the variable must be included) or "0" (if the variable must be excluded).

In our examples we wish to include in the output file:

- all 12 variables of record 01;
- all 17 variables of record 02;
- the first three, the eleventh and twelfth, and the fourteenth variables of record 03;
- finally, the first seven variables of record 04.

This ends the entry of the prefix lines requested by the COMPRESS program in front of an LDF.

(5) Next in our example, the user verifies the prefix lines and the first few lines of data before SAVEing the data file and ENDing this QED session.

```
READY
QED '       .PGMS.RUN(COMRECV)' OLD
ENTER PASSWORD FOR DATA SET       .PGMS.RUN:
:DATA SET NOT LINE NUMBERED-NONUM ASSUMED
QED
List
//SKCOMREC  JOB       ,MELLINGER,NOTIFY=      ,MSGCLASS=T
//COMRECV  EXEC PGM=COMRECV,PASS=
//STEPLIB DD UNIT=USER,DISP=SHR,DSN=       .COMREC.LOAD
//FT08F001 DD UNIT=USER,DISP=SHR,DSN=      .NWPT01.DATAIN
//FT09F001 DD UNIT=USER,DISP=(,CATLG),DSN=      .NWPT01.ASAM,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),SPACE=(6160,(100,10),RLSE)
//
END OF DATA
VERIFY
-3
//FT08F001 DD UNIT=USER,DISP=SHR,DSN=       .NWPT01.DATAIN
C /      .NWPT01.DATAIN/SKGSKR.QUIRT.NEWMERG/
//FT08F001 DD UNIT=USER,DISP=SHR,DSN=      .QUIRT.NEWMERG
C /NEWMERG/TEST.DATA/
//FT08F001 DD UNIT=USER,DISP=SHR,DSN=       .QUIRT.TEST.DATA
+
//FT09F001 DD UNIT=USER,DISP=(,CATLG),DSN=      .NWPT01.ASAM,
C /      .NWPT01.ASAM/      .QUIRT.COMPTEST/
//FT09F001 DD UNIT=USER,DISP=(,CATLG),DSN=      .QUIRT.COMPTEST,
List
//SKCOMREC  JOB       ,MELLINGER,NOTIFY=      ,MSGCLASS=T
//COMRECV  EXEC PGM=COMRECV,PASS=
//STEPLIB DD UNIT=USER,DISP=SHR,DSN=       .COMREC.LOAD
//FT08F001 DD UNIT=USER,DISP=SHR,DSN=       .QUIRT.TEST.DATA
//FT09F001 DD UNIT=USER,DISP=(,CATLG),DSN=      .QUIRT.COMPTEST,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),SPACE=(6160,(100,10),RLSE)
//
END OF DATA
SAVE
SAVED
```

(III.1. COMPRESS ... cont'd)

While in TSO it is usually necessary to modify the JCL which runs the COMPRESS program.

In our example, the user QEDits the file which contains this JCL, i.e., SKG1MM.PGMS.RUN (COMRECV). (Notes: this file is password-protected . . .please consult with the authors; also, the code name of the COMPRESS program is COMRECV, for COMpress RECords version V, because COMPRESS is a IBM system-reserved utility name.)

The user then edits the JCL file modifies the (1) input, and (2) output file names as required. After this, the JCL file is SAVEd (if required), and SUBmitted to run the program in batch mode.

The program will produce a compressed LDF under the file name given in //FT09F001 DD . . ., the output file Data Definition card. This compressed LDF is ready to be transferred to the Tektronix 4051, using the TRANSFER program (Section III.2).

NOTE: On the next page is a copy of the summarized instructions for the usage of the COMPRESS program.

```
00010 ********************************************************************************
00020 *                                                                              *
00030 **  COMRECV    (FORTRAN77)                                  *   INSTRUCTIONS   *
00040 *                                                                              *
00050 ********************************************************************************
00060 *
00070 **  FUNCTION  **
00080 *
00090 *       PROGRAM COMRECV IS USED TO COMPRESS A LITHOGEOCHEMICAL
00100 *       DATA FILE (LDF).
00110 *
00120 *       DESCRIPTIVE INFORMATION FROM RECORD 1 OF THE LDF IS
00130 *       RETAINED.  GEOCHEMICAL DATA FIELDS ARE SELECTABLE.
00140 *
00150 *
00160 **  REFERENCES  **
00170 *
00180 *       MELLINGER,M.,(1983).  THE LITHOGEOCHEMICAL DATA FILE,
00190 *       SRC PUB. R-740-1-E-83, 46 P.
00200 *
00210 *       SMITH,J.W.J., MELLINGER,M.,(1983).  GRAPHICS SOFTWARE
00220 *       FOR DATA ANALYSIS: II. DATA TRANSFER, DATA FILE
00230 *       MANAGEMENT - USER'S GUIDE.
00240 *
00250 *
00260 **  DIMENSION RESTRICTION  **
00270 *
00280 *       UP TO 100 GEOCHEMICAL DATA FIELDS PER SAMPLE ARE ALLOWED
00290 *
00300 *
00310 **  INPUT FILE **
00320 *
00330 *       UNIT 8 : LITHOGEOCHEMICAL DATA FILE  (LDF FORMAT)
00340 *
00350 *
00360 **  OUTPUT FILE  **
00370 *
00380 *       UNIT 7 : COMPRESSED LDF RECORDS
00390 *
00400 *
00410 **  CONTROL RECORDS  **
00420 *
00430 *       1 - TITLE - FORMAT A63:
00440 *           (SOME GRAPHICS PROGRAMS USE ONLY THE FIRST 30 CHARACTERS)
00450 *
00460 *       2 - NUMBER OF LDF RECORDS PER SAMPLE - FORMAT I4
00470 *
00480 *       3 - ADDITIONAL NAMES OF GEOCHEMICAL DATA FIELDS - FORMAT 17A4:
00490 *
00500 *           NAMES FOR THE FIRST THREE LDF RECORDS ARE GENERATED BY
00510 *           THE PROGRAM.  NEW NAMES NOT YET INCLUDED IN THE PROGRAM
00520 *           MUST BE ADDED.  EACH NAME CONSISTS OF 4 CHARACTERS
00530 *           AND MUST BE ENTERED IN THE SAME SEQUENCE AS THE DATA
00540 *           FIELDS IN THE LDF RECORD; SPACES MUST BE INCLUDED FOR
00550 *           UNUSED DATA FIELDS.  REPEAT 17A4 RECORD AS NEEDED.
00560 *
00570 *       4 - SELECTION CODES
00580 *           FIRST RECORD - FORMAT 12I1
00590 *           SUBSEQUENT RECORDS - FORMAT 17I1:
00600 *           CODE=1 - INCLUDE, CODE=0 - OMIT
00610 *           SELECTION CODES ARE SPECIFIED IN THE SAME ORDER
00620 *           AS THE OCCURRENCE OF GEOCHEMICAL DATA FIELD NAMES
00630 *
00640 *
00650 ********************************************************************************
00660 *
00670 **  JCL RECORDS  **
00680 *
00690 *   //JOBNAME   JOB   ACCOUNT,NAME,MSGCLASS=T,NOTIFY=TSOACCT
00700 *   //      EXEC PGM=COMRECV,PASS=PASSWORD
00710 *   //STEPLIB DD UNIT=USER,DISP=SHR,DSN=SKG100.COMREC.LOAD
00720 *   //FT06F001 DD SYSOUT=*
00730 *   //FT08F001 DD *
00740 *   //*  CONTROL RECORDS (EXAMPLE)                    (NOTES:)
00750 *   ATHABASCA GEOCHEMICAL DATA                       (TITLE)
00760 *      4                                             (NUMBER OF RECORDS)
00770 *   LA KADLCHLRILLI IR  AI CSFPANORMDEXCHOR          (ADDITIONAL NAMES)
00780 *   111111111111                                     (SELECTION CODES)
00790 *   1111111111111111                                    .
00800 *   11100000001101000                                   .
00810 *   11111110000000000                                   .
00820 *   //          DD UNIT=USER,DISP=SHR,DSN=ACCOUNT.LDF.DSNAME
00830 *   //FT09F001 DD UNIT=USER,DISP=(,CATLG),DSN=ACCOUNT.DSNAME,
00840 *   // DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),SPACE=(6160,(100,10),RLSE)
00850 *   //
00860 *
00870 *   NOTE:  //FT08F001 MAY POINT TO A DISK FILE CONTAINING THE
00880 *          CONTROL RECORDS
00890 *
00900 ********************************************************************************
00910 *
00920 *  INSTRUCTIONS WRITTEN JWS(APRIL 4/84)  **
00930 *
00940 ********************************************************************************
END OF DATA
```

```
TSO TO TEK 4051 DATA RECEIVE PROGRAM

PGM now entering TERMINAL MODE to allow user
to LOGON to TSO and QED the data file on the IBM
[*]
ENTER LOGON
[LOGON                ]
        LOGON IN PROGRESS AT 16:18:16 ON AUGUST 9, 1983
NO BROADCAST MESSAGES
READY
[QED QUIRT.COMP.NEWMERG OLD]
DATA SET NOT LINE NUMBERED-NONUM ASSUMED
QED
[RLN]
REL. LINE NUMBER IS 47

[User presses 'RETURN TO BASIC' key]

ENTER NUMBER OF RECORDS TO BE RECORDED [47]
SELECT RECEIVING FILE BY # [13]
MARKING FILE FOR DATA RECEPTION   3760 BYTES
NOW ENTERING TERMINAL MODE WITH TSO
[Data being transferred appear on screen]

DATA TRANSFER COMPLETED

REENTERING TERMINAL MODE: user should usually
END the QED session and LOGOFF TSO, then return to
BASIC program mode by pressing RETURN TO BASIC key
[END]
READY
[LOGOFF]

-------------------------------------------------------------------------
 CONNECT      TERMINAL      TERMINAL      DISK      CPU
TIME(MIN)      READS         WRITES        IO     TIME(SEC)
       2           5             53         3          1
-------------------------------------------------------------------------
        LOGGED OFF TSO AT 16:28:47 ON AUGUST 9, 1983
[User presses 'RETURN TO BASIC' key]
PGM ENDED
```

III.2 DATA TRANSFER PROGRAM

III.2.a Purpose: the data TRANSFER program is a Tektronix program which transfers the compressed data file from the IBM 4341 to the internal magnetic tape in the Tektronix 4051. It requires access to the compressed data file on the IBM 4341 under TSO/QED.

III.2.b Input: the compressed data file, located in a disk file on the IBM 4341. This file was created by the COMPRESS program (Section III.1).

III.2.c Output: an ASCII (character) data file on a magnetic tape in the Tektronix 4051.

III.2.d Program Operation: The user connects the Tektronix 4051 to the modem with the RS232 cable. The program COM-LINK is then loaded and RUN: the user-definable keys (UDK) overlay "DATA COMMUNICATION INTERFACE" is required.

(1) The PROGRAM informs the user that terminal mode is being entered.

(2) The USER LOGs ON to the TSO system.

(3) The USER accesses the compressed data file by QEDing 'filename'. The number of 80-character lines in this file is then requested by entering RLN (Relative Line Number); the computer replies (here: REL. LINE NUMBER IS 47).

(4) The USER returns the terminal to BASIC program mode by pressing the "RETURN TO BASIC" key (UDK #5).

(III. 2. TRASNFER ... cont'd)

(5) The PROGRAM requests the number of record to be recorded. The USER enters the number of records as determined in step 3 above (here equal to 47).

(6) The PROGRAM requests the file numbers of the receiving file (Tektronix internal tape).

(7) The PROGRAM marks the selected file on the Tektronix tape, returns to terminal mode, and initiates the transfer of data. The data records are displayed on the screen as they are transferred and the user can let the screen fill up until the end of the transfer.

(8) When the transfer is completed, terminal mode is re-entered so the user ENDs the QED session and LOGs OFF the TSO system.

(9) The TRANSFER program then ends, after the user presses the "RETURN TO BASIC" key (UDK #5).

The compressed data file is now on the Tektronix's magnetic tape file created in steps (6) and (7) above. It is an ASCII (character) file, and must be translated into a BINARY-coded file before further data processing on the Tektronix: this is accomplished by using the data TRANSLATION program (Section III.3).

```
RAM ASCII-BINARY TRANSLATOR
ENTER ASCII INPUT FILE # [5]
 1672812142MLECA81
IS THIS THE FILE YOU WANTED? Y/N [Y]
167 RECORDS: 28 NUMERIC VARIABLES
SAM1SAM2SAM3NROKCORXCORYCOR2FLAGUNITSUBUIROKMETATEXTFACSCL29CL30SIOXTIOX
ALOXCAOXKOX MGOXFE2XFE3XHOX COX NAOXPOX  MN  CU  ZN  PB  NI  LI  BA  RB
 SR  V   CO  AS  AG  U   MO  B
YOU MAY DESIGNATE SAM1, SAM2, SAM3 TO BE NUMERIC
FIELD MUST CONTAIN AT LEAST ONE VALID NUMERIC CHARACTER
ENTER SAM1, SAM2, SAM3 OR EXIT [EXIT]
VARIABLES 12 AND 13 ARE UNNAMED. NAME THEM? Y/N [N]
```

new screen

```
BINARY OUTPUT WILL BE STORED IN RAM
SPACE REQ'D= 70106 SPACE AVAILABLE= 276992
THE DIRECTORY WILL PRINT; SELECT AN UNUSED FILE#
NO.   FILE NAME        TYPE          REC LEN  USED BYTES
```

new screen

```
ENTER THE UNUSED FILE # FOR DATA STORAGE [120]
 1MLECA81!167 44 4 12 12 142
FIRST INPUT RECORD
10MLE0014910 10601 1 MS25  2  9700  20  70   5   7   1  55   2  23  18
 5   3  19   4   2   0   6  12  26  39  82   5   0 3.8 0.1   0   1  12
PROCESSING
```

new screen

```
LAST RECORD ID = 10MLE280
FILE COMPLETED
```

new screen

```
[CALL "DIR"]
NO.   FILE NAME        TYPE          REC LEN  USED BYTES
 .      .                .              .        .
 .      .                .              .        .
5120  MLECA81        SEQUENTIAL        8       69999
 .      .                .              .        .
 .      .                .              .        .
```

(1) (2) (3) (4) (5) (6) (7) (8)

III.3 DATA TRANSLATION PROGRAM

III.3.a Purpose: the data received from the IBM 4341 is in the form of ASCII characters and it must be translated into its binary equivalent prior to further data processing on the Tektronix. The data TRANSLATION program handles this task.

III.3.b Input: the ASCII (compressed) data file as received from the IBM 4341 (Section III.2), and located on a Tektronix magnetic tape.

III.3.c Output: on a magnetic tape a pair of binary files. The first file is a single-line header file which contains the title of the data set and parameters which describe the file contents of the second file, which contains the data proper. On a disk (floppy or RAM), these two files are concatenated into one by the TRANSLATION program.

III.3.d Program Operation: The version of the TRANSLATION program that saves the binary file on RAM-disk is RUN here as example. After identifying itself:

(1) The PROGRAM prompts the USER for the file # of the ASCII input file that has been created on the Tektronix magnetic tape (file #5 is used here), then displays the file header line and asks the user for confirmation.
The USER confirms ("Y") or not ("N") that the file accessed is the right one. If "N", the program re-starts. If "Y", the program proceeds:

(2) The PROGRAM displays the characteristics of the data file and its variable dictionary.

(III.3. TRANSLATION ... cont'd)

(3) Fields SAM1 (chars 1-2), SAM2 (chars 3-4), and SAM3 (chars 5-8) may be defined by the user as numeric or alphabetic fields, depending on how the 8-character sample number was structured. The PROGRAM prompts the USER to indicate which one(s) of the three fields will be defined as numeric fields; it will then re-order them accordingly in the binary file (note: field NROCK is always assumed alphabetic). In our example, the USER enters "EXIT" to indicate that none of SAM1 to SAM3 is numeric, i.e., all three fields must be treated as alphabetic.

(4) The PROGRAM allows the user to name the optional codes located in Col. 29 and Col. 30 of LDF record #01.
If the USER chooses to rename one or both coded variables ("Y"), then the program will request new names as appropriate. If the USER chooses to leave both coded variables unnamed ("N"), then the program proceeds and will keep CL29 and CL30 as default names. In our example, neither optional coded variable was used ("N").

(5) The PROGRAM indicates that the binary file will be stored on the RAM-disk and displays specifics. It then displays the RAM-disk directory (not shown here) and requests a file # for the binary file that will be produced and stored on the RAM-disk.
The USER selects a file # (file #120 is selected here).

(III.3. TRANSLATION ... cont'd)

(6) The PROGRAM proceeds with the ASCII-to-Binary translation, one record at a time: it first displays the file header line as well as the first record read; it then flashes "PROCESSING" on the screen until all records have been translated.

(7) After the translation is completed, the PROGRAM displays the sample number of the last record read for the information of the user, then displays the message "FILE COMPLETED" and ends.

(8) The USER may verify that the binary file has been created, by asking for the RAM-disk directory (key in: CALL "DIR", as shown).

In our example, file #120 is the binary file created, with the name "MLECA81" (note: the directory allows for up to 15 characters for the file title).

```
SCANSTAT3A - SCAN COMPRESSED BINARY DATA FILE FOR
MINIMUM, MAXIMUM, MEAN AND STD. DEVIATION FOR EACH VARIABLE
PROGRAM LISTS ONLY FIRST TEN ALPHAS ENCOUNTERED

OUTPUT IS TO DEC III: ENTER 'RETURN' WHEN READY

ENTER SUBJECT DATA FILE # 6

FILE TITLE: MLECA81

NO. SAMPLES= 167  NO. VARIABLES= 44

 1 SAM1    2 SAM2    3 SAM3    4 NROK    5 CORX    6 CORY    7 CORZ
 8 FLAG    9 UNIT   10 SUBU   11 IROK   12 META   13 TEXT   14 FACS
15 CL29   16 CL30   17 SIOX   18 TIOX   19 ALOX   20 CAOX   21 KOX
22 MGOX   23 FE2X   24 FE3X   25 HOX    26 COX    27 NAOX   28 POX
29  MN    30  CU    31  ZN    32  PB    33  NI    34  LI    35  BA
36  RB    37  SR    38  V     39  CO    40  AS    41  AG    42  U
43  MO    44  B

SCANNING DATA

PRINTING ON DEC III

PROGRAM ENDED
```

1

2

3

4

III.4 SCANSTAT PROGRAM

III.4.a Purpose: the SCANSTAT program provides the user with a preview of the data by displaying a partial list of the alphabetic variables present in the file and summary statistics for all numeric variables. This preview is printed on the printer terminal.

III.4.b Input: the binary data file produced by the TRANSLATION program (Section III.3) or by the SUBFILE program (Section III.6).

III.4.c Output: a printed report containing the list of alphabetic variables and summary statistics for the numeric variables (see example of output next page).

III.4.d Program Operation: The USER loads the SCANSTAT program into the Tektronix core memory, and RUNs the program:

(1) The PROGRAM displays messages for the user and requests the user to confirm the status of the (DEC III) printer, which must be powered ON and linked to the Tektronix. The USER presses the "RETURN" key when the printer is ready.

(2) The PROGRAM then requests the number of the binary file which contains the data to be examined. The USER enters the file number (here: file #6).

(3) The PROGRAM displays the file header, its size parameters, and its variable name list (dictionary).

(4) The PROGRAM then scans the data file and, when finished, prints the output on the DEC III printer. The program is then ended.

Note: The following page is an actual output from the SCANSTAT program.

HLECMS1 (1)

NUMBER OF SAMPLES= 167 NUMBER OF VARIABLES= 44 (2)

SAMPLE #, FIELD 1 10

SAMPLE #, FIELD 2 ML

SAMPLE #, FIELD 3 E001 E002 E003 E004 E005 E006 E007 E008 E009 E010 (3)

(4)

	VARIABLE	MINIMUM	MAXIMUM	MEAN	STD.DEVN.		VARIABLE	MINIMUM	MAXIMUM	MEAN	STD.DEVN.
5	CORX	49	57	-	-	25	HOX	0.01	9.08	1.26	1.76
6	CORY	10	40	-	-	26	COX	0.00	5.53	0.36	0.62
7	CORZ	1	119	-	-	27	NAOX	0.00	4.24	0.47	1.09
8	FLAG	1	1	-	-	28	POX	0.00	0.52	0.05	0.06
9	UNIT	1	2	-	-	29	MN	4	3750	133	420
10	SUBU	0	0	-	-	30	CU	3	440	12	36
11	IROK	3	93	-	-	31	ZN	2	180	16	24
12	META	0	0	-	-	32	PB	0	54	7	9
13	TEXT	0	0	-	-	33	NI	1	128	16	23
14	FACS	1	9	-	-	34	LI	2	494	45	58
11	CL29	-	-	-	-	35	BA	0	1403	111	223
12	CL30	-	-	-	-	36	RB	18	426	83	92
17	SIOX	45.30	98.50	84.95	13.69	37	SR	5	417	80	62
18	TIOX	0.01	1.80	0.36	0.34	38	V	0	297	33	49
19	ALOX	0.70	34.00	6.84	6.96	39	CO	0	48	6	9
20	CAOX	0.00	8.49	0.31	1.01	40	AS	0	11	3	2
21	KOX	0.02	8.06	1.31	2.00	41	AG	0	3	1	1
22	MGOX	0.00	12.95	1.01	1.96	42	U	0	14	1	2
23	FE2X	0.14	12.88	1.33	2.13	43	MO	0	31	4	3
24	FE3X	0.01	9.05	1.37	1.76	44	B	4	1001	99	208

PRINTED IN CANADA

(III.4. SCANSTAT PROGRAM ... cont'd)

Opposite is an example of the printout produced by the SCANSTAT program:

(1) the file title is printed;

(2) the file size is indicated;

(3) the character-fields are listed: all three fields from the sample number field (SAM1, SAM2, SAM3) are defined as character-fields for this file; the first 10 different occurrences of characters found in these three fields are listed (here, SAM1 and SAM2 contain the same characters for all samples);

(4) next, all numeric variables are listed and for each, summary statistics are printed: minimum value, maximum value, mean, standard deviation. The latter two are not calculated for the coded variables (#5 to 12).

```
CONTINGENCY TABLE - FROM COMPRESSED BINARY FILE
PROGRAM SCANS PAIRS OF COMPLEMENTARY CODES
ENTER SUBJECT DATA FILE # 6
FILE TITLE: MLECA81
NO. SAMPLES= 167  NO. VARIABLES= 44

 1 SAM1    2 SAM2    3 SAM3    4 NROK    5 CORX    6 CORY    7 CORZ
 8 FLAG    9 UNIT   10 SUBU   11 IROK   12 META   13 TEXT   14 FACS
15 CL29   16 CL30   17 SIOX   18 TIOX   19 ALOX   20 CAOX   21 KOX
22 NGOX   23 FE2X   24 FE3X   25 HOX    26 COX    27 NAOX   28 POX
29  MN    30  CU    31  ZN    32  PB    33  NI    34  LI    35  BA
36  RB    37  SR    38  V     39  CO    40  AS    41  AG    42  U
43  MO    44  B

CODED VARIABLES

 5 CORX    6 CORY    7 CORZ    8 FLAG    9 UNIT   10 SUBU   11 IROK   12 META
13 TEXT   14 FACS   15 CL29   16 CL30

SELECT CODE PAIR;(E.G. 1,5)  11,14

MLECA81

      FACS    1    2    3    4    5    8    9
-------------------------------------------------
IROK   3 |    0    0    0    0    2    0    0 |   2
      13 |    3    3    1    2    2    0    0 |  11

      21 |    2    0    0    0    0    1    0 |   3
      22 |    0    1    0    0    0    0    0 |   1
      23 |    1    0    0    0    0    0    0 |   1

      24 |   27    6    0    1    2    3    0 |  39
      25 |    9    6    1    0    2    2    0 |  20
      26 |   12   13    8    0    1    0    0 |  34

      70 |    1    0    0    0    0    0    0 |   1
      90 |    1    0    0    0    0    0    1 |   2
      91 |   11    5    1    0    0    0    1 |  18

      92 |   16    7    1    0    0    0    1 |  25
      93 |    1    5    3    0    0    0    1 |  10
-------------------------------------------------
         |   84   46   15    3    9    6    4 | 167

ANOTHER TABLE? Y/N
```

III.5 CONTINGENCY TABLE PROGRAM

III.5.a Purpose: the CONTINGENCY TABLE program produces contingency tables for pairs of descriptive variables. The user may thus examine both the range of values of the variables and their relationships.

III.5.b Input: the binary data file produced by the TRANSLATION program (Section III.3) or by the SUBFILE program (Section III.6).

III.5.c Output: contingency tables are written on the screen of the terminal and may be copied as required using the 4611 Hard Copy Unit.

III.5.d Program Operations: The USER loads the CONTINGENCY program into memory and RUNs it:

(1) The PROGRAM displays an introductory message for the user.

(2) The USER is requested to enter the file number of the binary data file to be examined. (File #6 will be examined here.)

(3) The PROGRAM reads the file header and lists information about the file.

(4) The PROGRAM then displays a list of the (coded) descriptive variables which may be examined. The user is requested to select a pair of variables by entering the index numbers.

The USER enters the variable numbers of interest (variables 11 and 14 will be examined here).

(5) The PROGRAM scans the data file and displays the contingency table obtained for the requested variables: #11

(III.5. CONTINGENCY TABLE ... cont'd)

(IROK) and #14 (FACS). A hard copy of the screen can be obtained now by pressing the "MAKE COPY" key.

(6) The PROGRAM then prompts the user for another contingency table. If the USER answers "Y" (yes), then the program will be re-initiated at Step (4). Otherwise ("N" = no), the program ends.

```
RAM SUBFILE CREATION PROGRAM

OPERATES ON LITHOGEOCHEM RAM DATA FILE
NOTE: 1 PERMANENT RAM FILE CREATED
      1 TEMPORARY RAM FILE CREATED AND DELETED

ENTER INPUT DATA FILE #120

MLECA81

SUBJECT FILE? Y/N Y
```

(1) (2)

new screen

```
MLECA81
 167 SAMPLES   44 VARIABLES

 1 SAM1   2 SAM2   3 SAM3   4 NROK   5 CORX   6 CORY   7 CORZ
 8 FLAG   9 UNIT  10 SUBU  11 IROK  12 META  13 TEXT  14 FACS
15 CL29  16 CL30  17 SIOX  18 TIOX  19 ALOX  20 CAOX  21 KOX
22 MGOX  23 FE2X  24 FE3X  25 HOX   26 COX   27 NAOX  28 POX
29  MN   30  CU   31  ZN   32  PB   33  NI   34  LI   35  BA
36  RB   37  SR   38  V    39  CO   40  AS   41  AG   42  U
43  MO   44  B
DO YOU WISH TO ALTER TITLE? Y/N Y
ENTER NEW TITLE; UP TO 72 CHARACTERS
MLE.SUB(FACS=1)
DO YOU WISH TO SELECT BY COLUMN? Y/N Y
ENTER COLUMN INDEX STRING.  E.G. 1,3,4-7,22,34.
NOTE THE TRAILING DECIMAL POINT
1-23.
DECODING INDEX STRING

YOU MAY SELECT ROWS OF DATA ON UP TO 5 COLUMN VARIABLES
0 BYPASSES SELECTION:  ENTER # OF SELECTION VARIABLES 1
ENTER VARIABLE INDEX 14
CHOOSE PARAMETER TYPE.  ENTER 1 OR 2
1 - DISCRETE, 2 - RANGE(MIN/MAX) 1
DISCRETE PARAMETER.  HOW MANY VALUES? 1
ENTER VALUE 1 1
```

(3) (4) (5) (6) (7)

new screen

```
RAM DIRECTORY:

NO.   FILE NAME          TYPE          REC LEN  USED BYTES
120   MLECA81            SEQUENTIAL    8        69999

WARNING: THE TEMPORARY FILE# IS THE # FOLLOWING THE WRITE
         FILE# SELECTED BELOW; BOTH FILES MUST BE AVAILABLE

ENTER WRITE FILE # 125

PROCESSING

CREATING PERMANENT RAM FILE

RECORDS RECORDED= 84

FILE COMPLETED
CALL "DIR"

NO.   FILE NAME          TYPE          REC LEN  USED BYTES
120   MLECA81            SEQUENTIAL    8        69999
125   MLE.SUB(FACS=1)    SEQUENTIAL    8        21905
```

(8) (9) (10) (11)

III SUBFILE PROGRAM

III.6.a Purpose: the SUBFILE program is used to select a subset of any binary data file; a data file can be visualized as a data table: rows (samples) *versus* columns (variables). Selection can be done on this data table on the basis of the columns and/or the rows. Columns (variables) selection is done by specifying column numbers, whereas rows (samples) selection is controlled by up to five criteria related to specified columns.

III.6.b Input: a binary data file produced by the TRANSLATION program (Section III.3) or by this program.

III.6.c Output: a new binary file of the same type as the files created by the TRANSLATION program (note: see WARNING in step (8)).

III.6.d Program Operation: The USER loads the SUBFILE program into memory and RUNS it:

(1) The PROGRAM displays introductory messages and requests the input file number.

The USER enters the input file number (here: #120).

(2) The PROGRAM then displays the file title and prompts the user to confirm the file choice before proceeding. The USER confirms by entering "Y" (yes), or can change the input file number by entering "N" (no) and having the program re-initiated at step (1). ("Y" was entered here).

(3) After confirmation of the input file number, the PROGRAM displays the file title, its characteristics, and a list of the variable index numbers and names.

(III.6. SUBFILE ... cont'd)

(4) The PROGRAM gives the opportunity to the user to give the subfile to be created a name that differs from the initial file name.

If the USER accepts by entering "Y" (yes), then the PROGRAM prompts the user for the name of the subfile to be created. The USER enters the file name (here: "MLE.SUB (FACS=1)"). Note: even though the program accepts 72 characters for the file title, it is recommended to make the first 15 characters unique and specific for the file, due to display limitations in other components of the software.

(5) The PROGRAM asks the user if column (variable) selection is required. If the USER answers "Y" (yes), as done in our example, then:

the PROGRAM prompts the user for a string that will define the column numbers which must be kept. An example of the syntax rules is also displayed: either single column numbers or ranges of successive columns can be entered; successive items must be separated by a comma; the column selection must end with a period. In our example, all variables (columns) from number 1 to 28 will be kept (the selection string is thus: "1-28.").

Next, the PROGRAM displays the message "DECODING INDEX STRING" to indicate that it is unravelling the column selection string just entered.

(III.6. SUBFILE ... cont'd)

(6) The PROGRAM now requests specifics for row (sample) selection. Criteria on up to 5 columns (variables) can be set for row selection, and the PROGRAM first prompts the user for the number of columns or selection variables that will be used.

The USER enters the number of selection variables required, or "0" if none is required. In our example, only 1 selection variable will be used.

(7) This step will be REPEATED for each selection variables required (here only 1 has been required).

The PROGRAM prompts the user for the index of the selection variable; the USER enters this index (here, variable #14, FACS, will be used).

The PROGRAM then prompts the user for the type of selection criterion to be applied: either discrete (code=1) or of range-type (code=2); the USER chooses the relevant option (discrete selection values will be used here).

The PROGRAM confirms the user's choice and prompts the user for the number of discrete selection values that will be used. The USER enters this number (only 1 discrete value will be used here).

The PROGRAM finally prompts the user for as many discrete values as were requested just above. The USER enters the discrete values successively (in our example, value #1, the only one requested, is entered as equal to 1).

(III.6. SUBFILE ... cont'd)

Note: if the selection criterion involves values of the range-type (criterion option = 2), then the PROGRAM prompts the user in similar fashion for a series of ranges of values for each selection variable.

(8) The PROGRAM can now proceed with the creation of the subfile according to the criteria defined by the user. In our example, the subfile will include:

- only columns (variables) 1 to 28 (SAM 1 to POX),
- and only those samples which have FACS = 1 as code value for the petrographic facies.

The PROGRAM displays the RAM-disk directory so that the user may choose a file # under which the subfile to be created will be saved. The USER enters the file # selected (here: file #125 ... note the WARNING: file #126 in our case must also be available, as temporary file used by the program).

(9) The PROGRAM flashes "PROCESSING" on the screen until the subfile creation is completed.

(10) The PROGRAM then displays appropriate messages for the user: the subfile is stored, the number of sample records which passed the selection criteria is displayed (here: 84 records passed), and the program ends.

(11) The USER can verify that the subfile has been stored properly on the RAM-disk by keying in CALL "DIR". The RAM-disk directory here shows the initial file (#120: MLECA81) and the subfile just created (#125: MLE.SUB(FACS=1)).

```
ALE.SUB(FACS=1)
NUMBER OF SAMPLES=   84   NUMBER OF VARIABLES=  28
SAMPLE #: FIELD 1  10
SAMPLE #: FIELD 2  ML
SAMPLE #: FIELD 3  E002 E004 E005 E007 E009 E011 E012 E014 E016 E017
```

	VARIABLE	MINIMUM	MAXIMUM	MEAN	STD.DEVN.		VARIABLE	MINIMUM	MAXIMUM	MEAN	STD.DEVN.
5	CORX	49	57	-	-	17	SIOX	46.39	97.69	84.72	14.61
6	CORY	10	40	-	-	18	TIOX	0.01	1.49	0.35	0.33
7	CORZ	4	114	-	-	19	ALOX	0.80	24.60	6.90	6.50
8	FLAG	1	1	-	-	20	CAOX	0.00	8.49	0.54	1.38
9	UNIT	1	2	-	-	21	KOX	0.02	7.19	1.73	2.31
10	SUBU	0	0	-	-	22	MGOX	0.00	8.63	0.95	1.82
11	IROK	13	93	-	-	23	FE2X	0.14	12.88	1.60	2.65
12	META	0	0	-	-	24	FE3X	0.01	5.18	0.22	0.89
13	TEXT	0	0	-	-	25	MOX	0.02	5.73	0.72	0.77
14	FACS	1	1	-	-	26	COX	0.00	5.53	0.38	0.82
15	CL29	-	-	-	-	27	NAOX	0.00	4.24	0.86	1.43
16	CL30	-	-	-	-	28	POX	0.00	0.52	0.05	0.07

(III.6. SUBFILE ... cont'd)

III.6.e Example of Subfile Created: The characteristics of the subfile created using the criteria shown in steps (5) to (7) of paragraph III.6.d, are shown here as a SCANSTAT printout for this subfile. Note the list of variables now present, as well as the range of values for variable FACS (compare with the printout in Section III.4, paragraph d).